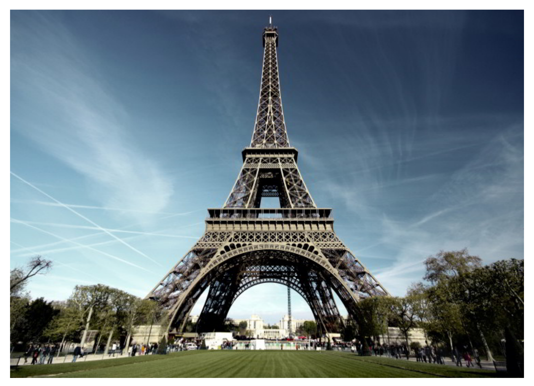
A Framework for Using Custom Features to Colorize Grayscale Images

Daniel J. Marchese
Advisor: James W. Davis
The Ohio State University

What is Image Colorization?

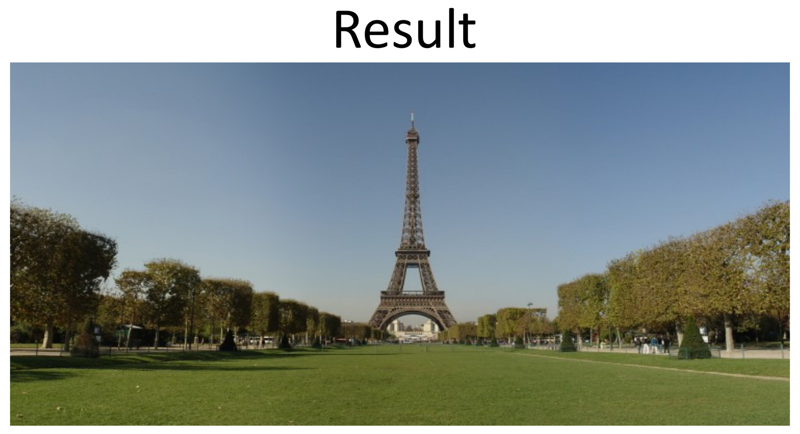
- The recovery of color context from a black-and-white image
  - Ill-posed (recovering 3 values from 1)
  - Composition of contexts
  - Learning Problem



Source

Target

Result



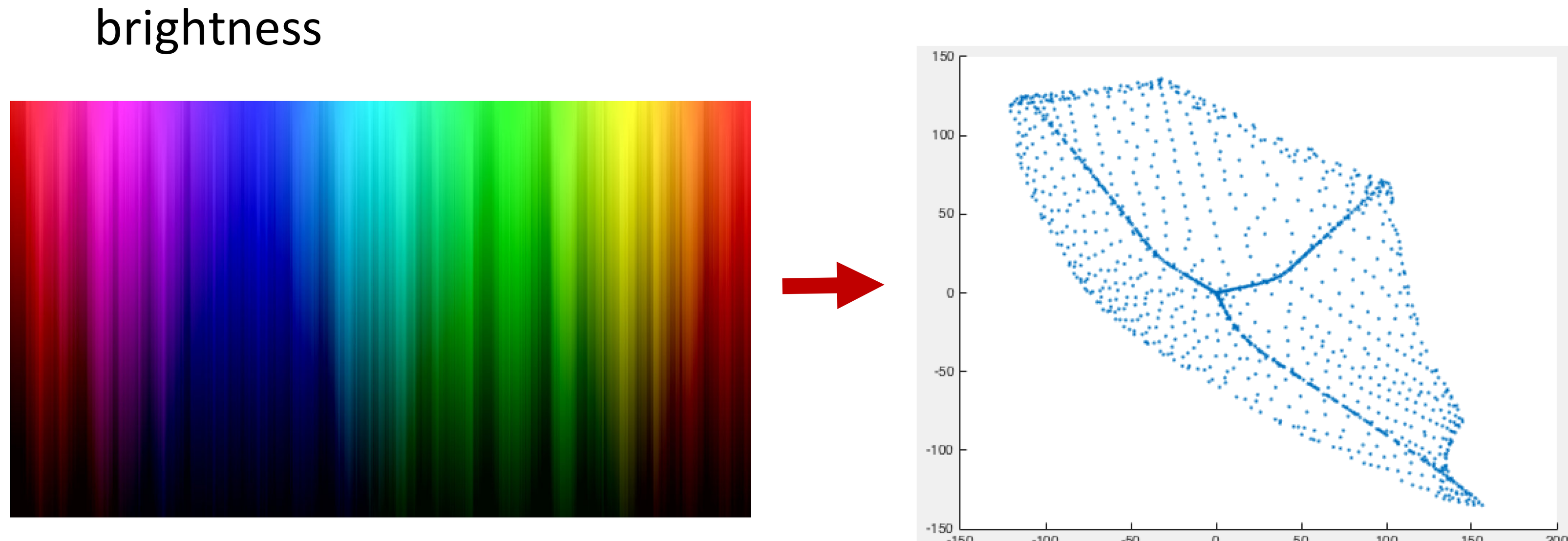
Framework Overview

- Calculate Image Features
  - Luminance, Variance, and 2-Vector
- Discretize the Color Space
  - Recover color palette
- Color Cost Mapping
  - Determine color labelling likelihoods
  - Formulate cost function
- Final Colorization
  - Solve for most likely coloring using cost function

Background

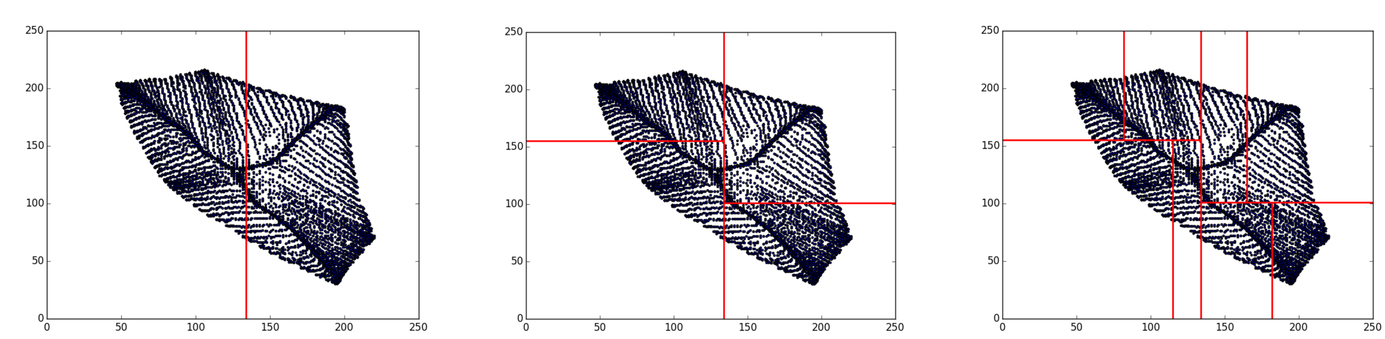
- SLIC Superpixels
  - Image Segmentation Technique based on K-Means
  - Improve overall algorithm performance

- The $L\alpha\beta$ color space
  - Separate Luminance and Chrominance
  - 1 less value to recover compared to RGB
  - Change color without changing brightness



Color Space Discretization

- Need to pick a color palette
  - Must capture the overall color "mood" of the scene
  - Must capture common yet subtle color variations
- Palette selections are based on density
  - High density regions get split in half along $\alpha\beta$ axes until reaching desired number of colors.



Color Cost Mapping

- Define the cost of assigning a color to a specific pixel
- Two Goals
  - Reward pixel region matches that are very close
  - Reward large number of matches in the same bin
- Formulate a cost function:
- Calculate a match score
  - Single Dimensional Case

$$S(c|p) = \sum_{i\in I} \frac{\left|\{q : q \in H_{i,b(p)} \wedge c = b_c(q)\}\right|}{\left|H_{i,b(p)}\right|}$$

  - Multi-Dimensional Case

$$S(c|p) = \sum_{i\in I} \left(\begin{cases} \dfrac{\left|M_{i,b(c)}\right|^2}{\sum_m^{M_{i,b(c)}} \|\boldsymbol{v}(m) - \boldsymbol{v}(p)\|}, & if \left|M_{i,b(c)}\right| > 0 \\ 0, & otherwise \end{cases} \right)$$

- Normalize the match scores
  - Single Dimensional Case

$$N(c|p) = \frac{S(c|p)}{\sum_{d\in C} S(d|p)}$$

  - Multi Dimensional Case

$$N(c|p) = \frac{S(c|p)}{\max_{c,p} S(c|p)}$$

- Formulate the color cost function

$$C(c|p) = 1 - N(c|p)$$

Final Colorization

- Need to preserve two things:
  - Local Coherence: Do my color selections make sense with my cost function?
  - Global Coherence: Do my color make sense in large contiguous pixel regions?
- Can be formulated as an energy:

$$E(L) = \sum_{p\in T} \left[\underbrace{\boxed{C(L_p|p)}}_{\text{Local Coherence}} + \lambda \underbrace{\boxed{\sum_{q\in n(p)} \frac{\|L_p - L_q\|_{L\alpha\beta}}{\max_{b_i,b_j\in B} \|b_i - b_j\|_{L\alpha\beta}}}}_{\text{Global Coherence}} \right]$$

- Can be solved for minimum energy efficiently using graph cuts

Evaluation Methods

- Qualitative
  - Do selected regions appear as expected?
  - Is the colorization *crisp*?
  - Do all of the colors make sense in the scene?
- Quantitative
  - Average Per-Pixel $L\alpha\beta$ distance

$$R(L) = \frac{\max_{b_i,b_j\in B} \|b_i - b_j\|_{L\alpha\beta} \sum \|L_p - C(q)\|_{L\alpha\beta}}{|T|}$$

Conclusion and Future Work

- Colorization results tend to be reasonable, however they are not quite up to par with state-of-the-art methodologies.
- Can tune the customization points better to achieve more reasonable results.
- Can re-formulate energy function to handle regions of expected color variance.